

LAB 5 – TASK 10

Linked List Student Database

John Dempsey

COMP-232 Programming Languages
California State University, Channel Islands

February 19, 2025

Hard Due Date: February 26, 2025

In this assignment, we will create a linked list student database.

To start the database application, users can type **db** and the **db>** prompt will be display. For example:

```
% db  
db>
```

The following commands will be supported:

add	Adds a student record to the database.
delete	Deletes a student record from the database.
list	Lists all current students in the database.
print	Prints information for one student.
help	Displays a helpful message on how to use the student database.
quit	Exits the student database.

EXTRA CREDIT: 3 points if you list the student's name in alphabetical order.

Users can also use just the first character to specify an operation, such as 'a' for add and 'd' for delete. Single letter commands are: a, d, l, p, h, and q.

Each of the above commands will briefly discussed.

add

To add a student to the database, the command is:

```
db> add <student name>,<age>,<class status>,<password>
```

where:

<student name> - represents the student's name.

<age> - how old the student is.

<class status> - Current class status, e.g., Freshman, Sophomore, Junior, or Senior.

<password> - Student's password.

An example is:

```
db> add Brian,20,Junior,helloworld
```

delete

To delete a student in the database, the command is:

```
db> delete <name>
```

For example,

```
db> delete Dalton
```

will delete student's Taylor entry in the database.

list

List will display a one line entry for all students in the database by typing:

```
db> list
```

The format will be:

```
Name: <student name>   Age: <age>   Class: <class>   Password: <password>
```

For 2 EXTRA CREDIT points: Sort the linked list of student names when the list command is run.

print

Prints information for one student in the following format. For example, print Clio displays:

```
db> print Lars
Print record for Lars:
Name      : Lars
Age       : 20
Year      : Senior
Password: iphone123
```

help

Help displays a helpful message for the user on how to use the database:

```
db> help
Welcome to COMP-232 Student Database
Valid commands are:
  add <name>,<age>,<class>,<password>
  where:
    name is student's name.
    age is student's age.
    year is what year the student is in.
    password is students's password.
  delete <name> - Deletes name from database.
  list - Lists all records in database.
  print <name> - Prints values for name.
  quit - Exits program.
  help - Displays this helpful message.
```

Additional Information

The database can handle any number of students, be implemented as a linked list, and memory must be allocated when adding a new student node using malloc or calloc.

When a student is deleted, the allocated memory for the student must be freed with a print statement indicating such.

When the database user quits the db application, all students and associated memory must be freed with a print statement indicating such for each student.

Error Messages

Helpful error messages need to be displayed when:

1. A user attempts to add a new student but forgets to enter the student's name, age, class status, or password.
2. A user attempts to delete a student from the database, but forgets to provide a name of the student to delete.
3. A user attempts to print information on a student, but forgets to provide a name of the student to print.

Sample Run

User inputs are bolded.

```
john@oho:~/LAB5.CODE$ db
```

```
db> add Imran,20,Junior,helloworld
```

```
Add record:
```

```
Name      : Imran
Age       : 20
Year      : Junior
Password: helloworld
```

```
-----
db> add Angelica,19,Junior,csuci
```

```
Add record:
```

```
Name      : Angelica
Age       : 19
Year      : Junior
Password: csuci
```

```
-----
db> add Alexander,20,Senior,camarillo
```

```
Add record:
```

```
Name      : Alexander
Age       : 20
Year      : Senior
Password: camarillo
```

```
-----
db> list
```

```
List all records:
```

```
1. Name: Imran           Age: 20, Year: Junior, Password: helloworld
2. Name: Angelica       Age: 19, Year: Junior, Password: csuci
3. Name: Alexander      Age: 20, Year: Senior, Password: camarillo
```

```
End list records.
```

```
-----
db> add Cyrus,19,Junior,ventura
```

```
Add record:
```

```
Name      : Cyrus
Age       : 19
Year      : Junior
Password: ventura
```

```
-----
db> add Bradley,20,Senior,iphone123
```

```
Add record:
```

```
Name      : Bradley
Age       : 20
Year      : Senior
Password: iphone123
```

db> list

List all records:

1. Name: Imran	Age: 20, Year:	Junior, Password: helloworld
2. Name: Angelica	Age: 19, Year:	Junior, Password: csuci
3. Name: Alexander	Age: 20, Year:	Senior, Password: camarillo
4. Name: Cyrus	Age: 19, Year:	Junior, Password: ventura
5. Name: Bradley	Age: 20, Year:	Senior, Password: iphone123

End list records.

db> delete Cyrus

Delete Cyrus:

Record Cyrus deleted.

Freeing space for record Cyrus

db> d Imran

Delete Imran:

Record Imran deleted.

Freeing space for record Imran.

db> list

List all records:

1. Name: Angelica	Age: 19, Year:	Junior, Password: csuci
2. Name: Alexander	Age: 20, Year:	Senior, Password: camarillo
3. Name: Bradley	Age: 20, Year:	Senior, Password: iphone123

End list records.

db> add Andrew,20,senior,channel*777

Add record:

Name : Andrew
Age : 20
Year : Senior
Password: channel*777

db> add Noemy,20,Senior,chicago654

Add record:

Name : Noemy
Age : 20
Year : Senior
Password: chicago654

db> list

List all records:

1. Name: Angelica	Age: 19, Year:	Junior, Password: csuci
2. Name: Alexander	Age: 20, Year:	Senior, Password: camarillo
3. Name: Bradley	Age: 20, Year:	Senior, Password: iphone123
4. Name: Andrew	Age: 20, Year:	Senior, Password: channel*777
5. Name: Noemy	Age: 20, Year:	Senior, Password: chicago654

End list records.

db> print Noemy

Print record for Noemy:

Name : Noemy
Age : 20
Year : Senior
Password: chicago654

db> print Angelica

Print record for Angelica:

Name : Angelica
Age : 19
Year : Junior
Password: csuci

db> list

List all records:

1. Name: Angelica	Age: 19, Year:	Junior, Password: csuci
2. Name: Alexander	Age: 20, Year:	Senior, Password: camarillo
3. Name: Bradley	Age: 20, Year:	Senior, Password: iphone123
4. Name: Andrew	Age: 20, Year:	Senior, Password: channel*777
5. Name: Noemy	Age: 20, Year:	Senior, Password: chicago654

End list records.

db> add

Add record:

Command: add. Missing student's name.

db> add Harrison

Add record:

Command: add Jack. Missing student's age value.

db> add Harrison,20

Add record:

Command: add Jack,20. Missing student's year.

db> add Harrison,20,Senior

Add record:

Command: add Harrison,20,Senior. Missing student's password.

db> delete

Missing name value to delete.

db> delete Kenna

Delete Kenna:

Kenna is not found in database.

db> print

Missing name value to print.

db> print Jason

Print record for Mitch:

Jason is not found in database.

db> **help**

Welcome to COMP-232 Student Database

Valid commands are:

add <name>,<age>,<class>,<password>

where:

name is student's name.

age is student's age.

year is what year the student is in.

password is students's password.

delete <name> - Deletes name from database.

list - Lists all records in database.

print <name> - Prints values for name.

quit - Exits program.

help - Displays this helpful message.

db> **q**

Free all allocated space.

Freeing space for record Angelica.

Freeing space for record Alexander.

Freeing space for record Bradley.

Freeing space for record Andrew.

Freeing space for record Noemy.

Exiting COMP232 Database.

Have a nice day.

john@oho:~/LAB5.CODE\$

As shown below, you can also run db with an input file containing commands too.

```
john@oho:~/LAB5.CODE$ ls  
SAVE db db.c input.txt run
```

```
john@oho:~/LAB5.CODE$ cat input.txt  
add Taylor,20,Junior,helloworld  
add Hayden,19,Junior,csuci  
add Evan,20,Senior,camarillo  
list  
add Jacob,19,Junior,ventura  
add Clio,20,Senior,iphone123  
list  
delete Taylor  
d Evan  
list  
add Frances,20,Senior,channel*777  
add Jerin,20,Senior,chicago654  
list  
print Clio  
print Frances  
list  
-- Now test error conditions.  
add  
add Jack  
add Jack,20  
add Jack,20,Senior  
delete  
-- Jorge doesn't exist in list.  
delete Jorge  
print  
-- Mitch doesn't exist in list.  
print Mitch  
-- Display Help Message.  
h  
quit
```


john@oho:~/LAB5.CODE\$ db < input.txt

db> Add record:

Name : Taylor

Age : 20

Year : Junior

Password: helloworld

db> Add record:

Name : Hayden

Age : 19

Year : Junior

Password: csuci

db> Add record:

Name : Evan

Age : 20

Year : Senior

Password: camarillo

db> List all records:

1. Name: Taylor	Age: 20, Year:	Junior, Password: helloworld
2. Name: Hayden	Age: 19, Year:	Junior, Password: csuci
3. Name: Evan	Age: 20, Year:	Senior, Password: camarillo

End list records.

db> Add record:

Name : Jacob

Age : 19

Year : Junior

Password: ventura

db> Add record:

Name : Clio

Age : 20

Year : Senior

Password: iphone123

db> List all records:

1. Name: Taylor	Age: 20, Year:	Junior, Password: helloworld
2. Name: Hayden	Age: 19, Year:	Junior, Password: csuci
3. Name: Evan	Age: 20, Year:	Senior, Password: camarillo
4. Name: Jacob	Age: 19, Year:	Junior, Password: ventura
5. Name: Clio	Age: 20, Year:	Senior, Password: iphone123

End list records.

db> Delete Taylor:

Record Taylor deleted.
Freeing space for record Taylor

db> Delete Evan:
Record Evan deleted.
Freeing space for record Evan.

db> List all records:
1. Name: Hayden Age: 19, Year: Junior, Password: csuci
2. Name: Jacob Age: 19, Year: Junior, Password: ventura
3. Name: Clio Age: 20, Year: Senior, Password: iphone123
End list records.

db> Add record:
Name : Frances
Age : 20
Year : Senior
Password: channel*777

db> Add record:
Name : Jerin
Age : 20
Year : Senior
Password: chicago654

db> List all records:
1. Name: Hayden Age: 19, Year: Junior, Password: csuci
2. Name: Jacob Age: 19, Year: Junior, Password: ventura
3. Name: Clio Age: 20, Year: Senior, Password: iphone123
4. Name: Frances Age: 20, Year: Senior, Password: channel*777
5. Name: Jerin Age: 20, Year: Senior, Password: chicago654
End list records.

db> Print record for Clio:
Name : Clio
Age : 20
Year : Senior
Password: iphone123

db> Print record for Frances:
Name : Frances
Age : 20
Year : Senior
Password: channel*777

db> List all records:

```
1. Name: Hayden           Age: 19, Year:      Junior, Password: csuci
2. Name: Jacob            Age: 19, Year:      Junior, Password: ventura
3. Name: Clio             Age: 20, Year:      Senior, Password: iphone123
4. Name: Frances         Age: 20, Year:      Senior, Password: channel*777
5. Name: Jerin            Age: 20, Year:      Senior, Password: chicago654
```

End list records.

db> -- Now test error conditions.

db> Add record:

Command: add. Missing student's name.

db> Add record:

Command: add Jack. Missing student's age value.

db> Add record:

Command: add Jack,20. Missing student's year.

db> Add record:

Command: add Jack,20,Senior. Missing student's password.

db> Missing name value to delete.

db> -- Jorge doesn't exist in list.

db> Delete Jorge:

Jorge is not found in database.

db> Missing name value to print.

db> -- Mitch doesn't exist in list.

db> Print record for Mitch:

Mitch is not found in database.

db> -- Display Help Message.

db> Welcome to COMP-232 Student Database

Valid commands are:

add <name>,<age>,<class>,<password>

where:

name is student's name.

age is student's age.

year is what year the student is in.

password is students's password.

delete <name> - Deletes name from database.

list - Lists all records in database.

print <name> - Prints values for name.

quit - Exits program.

help - Displays this helpful message.

```
db> Free all allocated space.  
Freeing space for record Hayden.  
Freeing space for record Jacob.  
Freeing space for record Clio.  
Freeing space for record Frances.  
Freeing space for record Jerin.  
Exiting COMP232 Database.  
Have a nice day.
```

```
-----  
john@oho:~/LAB5.CODE$
```